Random reading notes

2020

Reading notes (work only)[1]

[1] Internal links (PDFs and other Org files) will likely not work outside this computer.

A general modular framework for gene set enrichment analysis (ackermann-2009-general-mod)

Based on topGO package, three types of enrichment tests depending on the data:

- Tests based on *gene counts*: most popular family of tests, only requires the presence of a list of interesting genes; e.g., Fisher's exact test or binomial tests. See draghici-2006-babel.
- Tests based on *gene scores* or gene ranks; e.g., Kolgomorov-Smirnov like tests (GSEA) or Gentleman's Category, t-test, etc. This paper.

High variability of mitochondrial gene order among fungi (aguileta-2014-high-variab)

Analysis of 38 complete fungal MT genomes, including Podospora Anserina. The authors conclude that unlike the case of Metazoa there exists MT recombination in all fungal phyla, arising as a result of nonhomologous and intrachromosomal recombination, sequence repeats at intergenic region and probably mobile elements dynamic.

Phylogenetic analysis: Protein sequences of the orthologs shared by all sampled species were aligned using a combi- nation of six different alignment strategies, and these alignments were automatically trimmed with trimAl, and then concatenated. Phylogenetic reconstruction was performed using PhyML and RAxML. Evolutionary rates were estimated with the r8s software.

Genome structural variation discovery and genotyping (alkan-2011-genom)

Review of existing technologies and introduction to NGS analysis. Useful glossary.

OMA standalone: orthology inference among public and custom genomes and transcriptomes (altenhoff-2019-oma)

- Orthology resources: eggNOG, Ensembl Compara, InParanoid, MBGD, OrthoDB, OrthoMCL, PANTHER, PhylomeDB, and OMA.
- OMA standalone app, available *via* Homebrew.

Orthologous and paralogous genes are two types of homologous genes, that is, genes that arise from a common DNA ancestral sequence. Orthologous genes diverged after a speciation event, while paralogous genes diverge from one another within a species. Put another way, the terms orthologous and paralogous describe the relationships between genetic sequence divergence and gene products associated with speciation or genetic duplication. (The difference between orthologous & paralogous genes)

Count-based differential expression analysis of rna sequencing data using r and bioconductor (anders-2013-count-rna)

De facto standard pipeline for RNA-Seq analysis using `TopHat` + `HTSeq` + `DESeq2`. See also kim-2019-graph-hisat for the successor of `TopHat2`.

See also conesa-2016-survey-best for a review of current best pratices and alternative workflows.

Note that `DESeq2` and `edgeR` use different defaults: Regarding *normalization*, edgeR uses the trimmed mean of M values while DESeq relies on a virtual reference sample; dispersion estimates are based on a trended mean in edgeR, whereas DESeq takes the maximum of the individual dispersion estimates and the dispersion-mean trend.

Improvement in protein domain identification is reached by breaking consensus, with the agreement of many profiles and domain co-occurrence (bernardes-2016-improv-protein)

`http://www.lcqb.upmc.fr/CLADE/`

A mathematical approach to studying fungal mycelia (boswell-2003-mathem-approac)

The model connects physiology at the hyphal level (e.g. tip growth and branching) to growth and function at the mycelial level.

- change in active hyphae in a given area -> new hyphae (laid down by moving tips) + reactivation of inactive hyphae – inactivation of active hyphae
- change in inactive hyphae in a given area -> inactivation of active hyphae – reactivation of inactive hyphae – degradation of inactive hyphae
- change in hyphal tips in a given area -> tip movement out of / into area + branching from active hyphae – anastomosis of tips

into hyphae

- change in internal substrate in a given area -> translocation (active and passive mechanisms) + uptake into the fungus from external sources – maintenance costs of hyphae – growth costs of hyphal tips – active translocation costs
- change in external substrate in a given area -> diffusion of external substrate out of / into area – uptake by fungus

See Fig 1 for an example of the expected power law for radial growth.* boswell-2012-model - Modelling hyphal networks Review of lattice-based and lattice-free network models.

- lattice-based models: essentially like cellular automata, discrete in time and space. The main limitation is that its topology is constrained by the grid or lattice.
- lattice-free models: mixture of deterministic and stochastic elements.; neighbour-sensing mathematical model.

Note: Hopkins and Boswell (2012) used a circular random walk to model tip orientation and related this to the corresponding Fokkere-Planck partial differential equation.

Many papers by Boswell on this topic.

Near-optimal probabilistic rna-seq quantification (bray-2016-near)

Easy to setup (`brew install kallisto`) and time+memory-efficient on fungi data.

Works on Galaxy server too. Beware that it returns different counts (TPM) than BEDtools multicov. See why: RPKM, FPKM and TPM, clearly explained and Counts vs. FPKMs in RNA-seq. See also this thread on SEQanswers.

A phylogenomics approach for selecting robust sets of phylogenetic markers (capella-gutierrez-2014)

Set of 4 genes in the case of ascomycetous fungal species (*Basidiomycota*):

YHR186C	1557	Target of rapamycin complex 1 subunit KOG1
YMR012W	1277	Clustered mitochondria protein 1
YJL029C	822	Vacuolar protein sorting-associated protein 53
YAR007C	621	Replication factor A protein 1

Phylogenetic tree analysis using PhyML, with Robinson and Foulds distance to compare trees. Interesting approach to use train/test dataset and resampling strategy.

Molecular Population Genetics (casillas-2017-molec-popul-genet)

Driving forces for *evolution*:

- natural selection: (ignoring effects of genetic drift) classical (homozygous loci for the wild-type allele) vs. balance (polymorphic loci) hypothesis, which requires to be able to estimate genetic diversity in populations. This has successively be done using allozyme polymorphisms (inconclusive results due to limitations of protein electrophoresis), nucleotide sequence data (using restriction enzymes, before PCR and automated Sanger sequencing), and genome variation.
- genetic drift,
- mutation,
- recombination,
- gene flux.

Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis (castresana-2000-selec-conser)

Instead of removing divergent regions in an arbitrary way, or use alternative approach that consist in assigning gap weights highly variable regions, the author proposes an algorithm (`GBlocks`) that accounts for: the degree of conservation of every position, stretches of contiguous nonconserved positions, minimum length support, removing all positions with gaps and nonconserved positions adjacent to them., as well as small block remaining after gap cleaning are also removed. The paper is quite old by now, and probably outdated.

Human genome assembly in 100 minutes (chin-2019-human-genom)

Long-read assembly, using an overlap-layout-consensus (OLC) paradigm, requires all-to-all read comparisons, which quadratically scales in computational complexity with the number of reads. Peregrine can assemble 30x human PacBio CCS read datasets in less than 30 CPU hours and around 100 wall-clock minutes to a high contiguity assembly (N50 > 20Mb).

A genome tree of life for the fungi kingdom (choi-2017-tree-life)

Gene tree (small number of highly conserved and orthologous genes) vs. genome tree (whole-genome DNA sequence, transcriptome RNA sequence, proteome amino acid sequence, exome DNA sequences, or other genomic features)

The authors rely on the whole-proteome sequences on the Feature Frequency Profile (FFP), which does not require multiple sequence alignment.

Mathematical Modelling Of Fungal Growth And Function (davidson-2011-mathem-model)

Summary of keynotes given at the SIG meeting on *Mathematical modelling of fungal growth and function*.

Graeme Boswell: discrete-continuous hybrid approach to modelling a fungal mycelium developing in a planar environment. Relies on Michael-Menten dynamics. See also boswell-2003-mathem-approac and boswell-2007-devel-fungal.

Star: ultrafast universal rna-seq aligner (dobin-2013-star)

STAR = Spliced Transcripts Alignment to a Reference

Designed to align the non-contiguous sequences directly to the reference genome, instead of short reads to a database of splice junctions or align split-read portions contiguously to a reference genome, or a combination thereof.

Algorithm: (1) MMP seed search and (2) clustering and stitching of all the seeds that were aligned to the genome (allowing for only one insertion or deletion) using local scoring scheme.

Orthofinder2: fast and accurate phylogenomic orthology analysis from gene sequences (emms-2018-orthof)

OrthoFinder infers orthogroups, genes trees, gene duplication events, the rooted species tree and extensive comparative genomic statistics. It has been shown to perform better compared to methods that use approximate phylogenetic relationships between genes using "reciprocal best hits" from BLAST (e.g., InParanoid, OrthoMCL and OMA).

Orthofinder provides accurate and scalable ortholog inference using gene trees in 3 stages: (1) orthogroup inference, (2) inference of rooted species and gene trees, and (3) inference of orthologs and gene duplication events from these rooted gene trees. Under the hood, it uses a duplication-loss-coalescent (DLC) resolution algorithm to identify gene duplication events and map them to the species tree.

Orthofinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy (emms-2015-orthof)

Two strategies: (1) inferring pairwise relationships between genes in two species, and then extending orthology to multiple species by identifying sets of genes spanning these species in which each gene- pair is an orthologue, (2) identify complete orthogroups; an orthogroup is the set of genes that are descended from a single gene in the last common ancestor of all the species being considered.

Fundamental biases in whole genome comparisons = Gene length bias in BLAST E-values affects the accuracy of orthogroup detection (fixed using normalization, p.9); over- or under-clustering of sequences (aka, phylogenetic distance from sequence similarity scores).

The structure of DNA (ferry-2019-dna)

Of historical importance only.

The space of tree-based phylogenetic networks (fischer-2019-space-tree)

Phylogenetic networks are trees with additional edges passing between the tree edges, that allow to account for horizontal gene transfer and hybridization.

Geometric approach to phylogenetic networks: consider the set of networks as a space in which one may move between the objects by operations that change a feature of the graph, e.g. nearest neighbor interchange (NNI), subtree prune and regraft (SPR) and tree bisection and reconnection (TBR).

19 dubious ways to compute the marginal likelihood of a phylogenetic tree topology (fourment-2018-dubious-ways)

The authors use the JC69 model to benchmark 19 methods for computing the marginla likelihood of a topology with respect to branch lengths. While the slowest, Generalized Stepping Stone (GSS) is the one that performs best. Gamma Laplus Importance Sampling (GLIS) is the best fast method, with performance wlose to GSS.

Recursive algorithms for phylogenetic tree counting (gavryushkina-2013-recur-algor)

In a Bayesian context,[2] this article describes a quadratic algorithm for counting the number of possible fully ranked trees on n sampled individuals (*aka* fully ranked tree with sampled ancestors).

[2] A general problem in evolutionary biology is how to reconstruct the phylogenetic tree from sequence data obtained from sampled individuals. Tackling this problem in a Bayesian framework may require counting the number of all possible histories on a sample of individuals.

Ranking top-k trees in tree-based phylogenetic networks (hayamizu-2019-rankin)

Support tree and linear-time algorithms for counting, enumeration and optimization (Hayamizu's structure theorem, arXiv:1811.05849).

Top-k ranking problem: list top-k support trees of N = (V,A) in non-increasing order by their likelihood values. This is a generalization of the top-1 ranking problem, which asks for a ML support tree of N

See also: Characterizing the Phylogenetic Tree-Search Problem.

Analysis of fungal networks (heaton-2012-analy-fungal-networ)

p.14 visualisation of network structure and network extraction

The network topology is defined by classifying junctions (branch points (degree 3), anastomoses and tips (degree 1)) as nodes and the chords between nodes as links. While the number of nodes and links increase over time, there's a selective loss of connections and thinning out of the fine mycellium. This shift can be quantified using the alpha coefficient, which gives the number of closed loops or cycles present as a fraction of the maximum possible for a planar network with the same number of nodes (Euler's polyhedral formula, V - E + F = 2).

The frequency distribution of node strength (i.e., summing the weight of all links connected to the node) shows more diversity than node degree alone, and follows an approximately log-normal distribution for *P. velutina* networks.

On the widespread and critical impact of systematic bias and batch effects in single-cell RNA-Seq data (hicks-2018-rna-seq)

> We found that the proportion of genes reported as expressed explains a substantial part of observed variability and that this quantity varies systematically across experimental batches. Furthermore, we found that the implemented experimental designs confounded outcomes of interest with batch effects, a design that can bring into question some of the conclusions of these studies.

Proposed experimental design (to control batch effects): account for differences in the proportion of detected genes by explicitly including the batch factor as a covariate in a linear regression model, while making use of biological replicates so that multiple batches of cells could be randomized across sequencing runs, flow cells and lanes as in bulk-RNA-Seq.

Growing scale-free networks with tunable clustering (holme-2002-growin)

Social networks, computer networks or metabolic networks have a logarithmically growing average geodesic (shortest path) length and an approximately algebraically decaying distribution of vertex degree.

The degree of an arbitrary vertex increases as the square root of the time, which yields the power-law degree distribution $P(k) \sim k^{-3}$.

See `networkx.powerlaw_cluster_graph`.

Ortholog-finder: a tool for constructing an ortholog data set (horiike-2016-orthol-finder)

Identifying genuine orthologs among distantly related species is challenging, because genes obtained through horizontal gene transfer (HGT) and out-paralogs derived from gene duplication before speciation are often present among the predicted orthologs.

This software uses 5 stages to alleviate such concern: (1) HGT filtering: Genes derived from HGT could be detected and deleted from the initial sequence data set by examining their base compositions. (2) Out-paralog filtering: Out-paralogs are detected and deleted from the data set based on sequence similarity. (3) Classification of phylogenetic trees: Phylogenetic trees generated for ortholog candidates are classified as monophyletic or polyphyletic trees. (4) Tree splitting: Polyphyletic trees are bisected to obtain monophyletic trees and remove HGT genes and out-paralogs. (5) Threshold changing: Out-paralogs are further excluded from the data set based on the difference in the similarity scores of genuine orthologs and out-paralogs.

Remark: See lechner-2014-orthol-detec for an intermediate approach (tolerate recent in-paralogs as unavoidable contamination).

Rna-seq analysis in mev (howe-2011-rna-seq-mev)

Latest standalone app dates back to 2011 and is Java 6 only. The Shell script included is useful for microarrays only.

Application of phylogenetic networks in evolutionary studies (huson-2006-applic-phylog)

Phylogenetic networks should be employed when *reticulate events* such as hybridization, horizontal gene transfer, recombination, or gene duplication and loss are believed to be involved.

Software: SplitsTree4.

- phylogenetic network = any network in which taxa are represented by nodes and their evolutionary relationships by edges.
- split network = combinatorial generalization of phylogenetic trees, designed to represent incompatibilities within and between data sets.
- reticulate network = represents evolutionary histories in the presence of reticulate events (nodes with two parents). (See Fig. 1 for an overview)

A split network contains exactly the same information as a list of splits with a weight for each split.

A survey of combinatorial methods for phylogenetic networks (huson-2011-survey-combin)

Phylogenetic networks are useful when evolution involves reticulate events (hybridization, horizontal gene transfer, or recombination) or to represent conflicts in a data set that may be caused by mechanisms such as incomplete lineage sorting.

Split networks and quasi-median networks are two examples of unrooted phylogenetic networks.

Sneath P. 1975. Cladistic representation of reticulate evolution. Syst Zool. 24(3):360–368.

Data-driven hypothesis weighting increases detection power in genome-scale multiple testing (ignatiadis-2016-data)

Independent hypothesis weighting (IHW): a method that assigns weights using covariates (conditionally) independent of the P-values under the null hypothesis but informative of each test's power or prior probability of the null hypothesis.

Deciphering the regulatory genome of escherichia coli, one hundred promoters at a time (ireland-2020-decip-escher)

Problem with modern biology is that although we have complete sequence of some important genomes, we know nothing about most of gene regulation (promoters).

First, we show that our method recapitulates regulatory information from known sequences. Then, we examine the regulatory architectures for more than 80 promoters in the E. coli genome which previously had no known regulation. In many cases, we also identify which transcription factors mediate their regulation.

Algorithms, data structures, and numerics for likelihood-based phylogenetic inference of huge trees (izquierdo-carrasco-2011-algor)

Design of a new search algorithm for large datasets: relies on a *backbone* tree, to reduce the dimensionality of the search space; basically, the idea is to collapse taxa that are closely related to each other into a single virtual tip. The virtual tips are then interpreted as tips in the backbone tree on which we can conduct the tree search. Optimal tree size reduction factor: R > 0.25.

Putting phylogeny into the analysis of biological traits: a methodological approach (jombart-2010-puttin)

Phylogenetic comparative methods (PIC, GLS, etc.) aim to correct for phylogeny (viewed as a nuisance factor) in the correlative analysis of biological traits at the species level.

The authors present a method which uses phylogenetic information to uncover the main phylogenetic structures observable in multivariate data associated with a phylogeny. Our approach, phylogenetic principal component analysis (pPCA), extends a methodology developed in spatial ecology (Dray et al., 2008) and in spatial genetics (Jombart et al., 2008) to the analysis of phylogenetic structures in biological features of taxa such as life-history traits.

- Dray, S., Saïd, S., Debias, F., 2008. Spatial ordination of vegetation data using a generalization of Wartenberg's multivariate spatial correlation. Journal of Vegetation Science 19, 45–56.
- Jombart, T., Devillard, S., Dufour, A.-B., Pontier, D., 2008. Revealing cryptic spatial patterns in genetic variability by a new multivariate method. Heredity 101, 92–103.

Identification of mammalian orthologs using local synteny (jun-2009-identmammal)

- differentiating between genes that have diverged through a speciation event (orthologs) and those derived through duplication events within a species (paralogs). Gene order may be viewed as a measure of conservation, or better gene family evolution.
- local synteny (gene order) might be useful to resolve ambiguous sequence based matches between putative orthologs (and retrogenes).
- 93% agreement between coding sequence based orthology (Inparanoid) and local synteny based orthology, with cases of discordance

resulting from evolutionary events including retrotransposition and genome rearrangements.

- intron conservation ratio = #(positional homologous introns)/#(intron positions in protein alignment), in strong agreement with the orthology assignments made by the two methods.

Model use in phylogenetics: nine key questions (kelchner-2006-model-use-phylog)

(1) What are models in phylogenetics; (2) Must a model be "exact" or merely "good enough"; (3) What phylogenetic applications rely on best-fit models; (4) What happens when a model is "wrong"; (5) How are models selected for nucleotide data; (6) What models are most frequently chosen for sequence data; (7) How can model selection methods be improved; (8) Are all parameters equally important; (9) Will phylogenomics eliminate the need for model selection.

Conceptual models often obeys to the principle of parsimony and they usually share several assumptions, that are not given formal parameters: mutations are independent and identically distributed, tree-like evolution (i.e., lineages arise in a divergent manner without reticulation), stationarity, reversibility, Markov process. Such assumptions are often violated in practice, e.g. prokaryote groups share genes among lineages via lateral gene transfer (incompatible with tree-like evolution).

Most complex model (10 parameters) = GTR+I+G (general time reversible model with corrections for invariant characters and gamma-distributed rate heterogeneity). See also `https://arxiv.org/abs/0709.0531v2`.

When models matter? Topology is quite robust to midly inadequate models, but when branch lengths matter or when we are interested in testing an alternative phylogenetic hypothesis (e.g., Kishino-Hasegawa, Shimodaira-Hasegawa and Incongruence Length Difference tests), we need more accurate and adequate models.

See also: goldman-2000-likel-based, shimodaira-2002-approx-unbias, darlu-2002-when-does.

How the strengths of lisp-family languages facilitate building complex and flexible bioinformatics applications (khomtchouk-2018-how-lisp)

See also the biolisp project and, e.g., herzeel-2015-elprep.

Rust-bio: a fast and safe bioinformatics library (koster-2016-rust-bio)

`https://rust-bio.github.io`

The net of life: reconstructing the microbial phylogenetic network (kunin-2005-net-life)

Horizontal Gene Transfer is viewed as a scale-free graph, allowing genes to propagate extremely rapidly across microbial species using certain organisms as hubs.

A bayesian mixture model for across-site heterogeneities in the amino-acid replacement process (lartillot-2004-bayes-mixtur)

The authors discuss a new model (CAT) for molecular phylogenetics which considers amino acids instead of nucleotides (which makes sense since they are interested in phylogenies going beyond the genus level). The model allows for a number of K classes, each of which is characterized by its own set of equilibrium frequencies, and lets each site "choose" the class under which its substitutional history is to be described. A Dirichlet process prior is used to decide on the best class to chose (the posterior mean then becomes a measure of substitutional heterogeneity). In sum, the CAT model allows to classifies sites into categories: sites are distributed according to a mixture of K distinct classes, each class being characterized by its own substitution matrix. Transition matrices (also called Pi-vector or *profiles*) can be fixed to pre-specified values (if all relative rates are set to 1, we get a Poisson process).

See also lartillot-2009-phylob. Maybe dang-2019-stoch-variat.

Rna-seq gene expression estimation with read mapping uncertainty (li-2010-rna-seq)

- Optimal read length = 20-25 bp.
- Problem with RMPKM measures: the mean expressed transcript length may vary between samples. (When the mean expressed transcript length is 1 kb, 1 TPM is equivalent to 1 RPKM, which corresponds to roughly one transcript per cell in mouse.)

Minimap2: pairwise alignment for nucleotide sequences (li-2018-minim)

Minimap2 is a general-purpose alignment program to map DNA or long mRNA sequences against a large reference database. It works with accurate short reads of 100 bp in length, 1 kb genomic reads at error rate 15%, full-length noisy Direct RNA or cDNA reads and assembly contigs or closely related full chromosomes of hundreds of megabases in length.

Used in the Eoulsan toolkit.

Rna-seq differential expression studies: more sequence or more replication? (liu-2014-rna)

Better to sequence less reads but increase the number of biological replicates: this will significantly increase the number of DE genes while the number of sequencing reads have a diminishing return after 10M reads.

Moderated estimation of fold change and dispersion for rna-seq data with deseq2 (love-2014-moder-rna-deseq)

NGS analyses (RNA, CHIP, etc.) need to account for within-group variance estimates when analysing lot of genes, hence the need to pool information across genes. The DESeq approach detects and corrects dispersion estimates that are too low through modeling of the dependence of the dispersion on the average expression strength over all samples. In addition, it provides a novel method for gene ranking and the visualization of stable estimates of effect sizes. The DESeq2 package further includes shrunken fold changes (with SE).

See also: ignatiadis-2016-data, zhu-2019-heavy, stephens-2017-false.

Simulating Colonial Growth Of Fungi With The Neighbour-Sensing Model Of Hyphal Growth (meskauskas-2004-simul-colon)

NS model = vector-based model whereby the growth vector of each virtual hyphal tip is calculated by reference to the surrounding virtual mycelium. It can be seen as an extension of stochastic L-system based approach.

This model can be used to simulate growth in semi-solid substrata like agar or soil, and it can be extended to include a number of other parameters and modelling capabilities that permit initial experimentation on hyphal growth kinetics, and enable realistic simulation of mycelial colonies of filamentous fungi grown in "Petri-dish style" experimental conditions.

High-quality sequence clustering guided by network topology and multiple alignment likelihood (miele-2012-high)

- Python 2.7 only
- Download: `http://lbbe.univ-lyon1.fr/Download,3100.html`

Statistical significance in biological sequence analysis (mitrophanov-2006-statis)

Review of sequence alignment score and the estimation of associated p-value in the case of single sequence studies (both score-based and score-free), global and local pairwise sequence alignments, multiple alignments, sequence-to-profile alignments and alignments built with hidden Markov models.

A biologist's guide to bayesian phylogenetic analysis (nascimento-2017-bayes)

Rules of thumb: (1) Different substitution models tend to give very similar sequence distance estimates when sequence divergence is less than 10%, so that a simple model can be used even though it may not fit the data. (2) It is more problematic to under-specify than to over-specify the model in Bayesian phylogenetics.

Inparanoid 7: new algorithms and tools for eukaryotic orthology analysis (ostlund-2010-inpar)

See also remm-2001-autom-clust.

A reference guide for tree analysis and visualization (pavlopoulos-2010-ref-tree)

Challenge: to handle the overload of information and make it easier to understand and explore very large phylogenetic trees.

- Trees vs. graphs.
- Cladogram and phylogram (branch lengths are proportional to the amount of inferred evolutionary change).
- Newick, NHX (enhanced Newick) and Nexus format.
- Statistical methods: neighbor-joining and UPGMA (distance), maximum parsimony and maximum likelihood (feature matrix), MCMC (both).

Reliable identification of genomic variants from RNA-Seq data (piskol-2013-reliab-ident)

Use `cufflinks` after `tophat2` for gene quantification.

RNA-seq data alone enabled the discovery of 40.2% and 47.7% of all coding variants identified by WGS in GM12878 cells and PBMCs, respectively. At the same time, RNA-seq only required a fraction (1/6) of the sequencing effort.

Tree disagreement: measuring and testing incongruence in phylogenies (planet-2006-tree-disag)

Review of incongruence tests for phylogenetic analysis: character information (character incongruence) vesus those that only consider tree shape or topology (topological incongruence); the latter presents the advantage of being able to compare trees derived from data that may not be strictly comparable or easy to include in the same analysis.

- **Character congruence:** incongruence length difference test, localized incongruence length difference, multiple partitions and pairwise ILD tests, ILD outside of parsimony, parsimony-based tests (permutation and sitewise tests), likelihood-based tests, sitewise testing, non-parametric bootstrapping methods, parametric bootstrapping and partition tests, Bayesian testing
- **Topological congruence:** consensus-based measurements, tree distances (symmetric difference)

Sequenceserver: a modern graphical user interface for custom blast databases (priyam-2019-sequen)

Only very basic sequence aligner. Not much compared to good old Wwwblast unfortunately. The only interest is possibly to use the automated converter of Fasta files (`makeblastdb`).

Automatic clustering of orthologs and in-paralogs from pairwise species comparisons (remm-2001-autom-clust)

Orthology analysis between humans and invertebrates is often complex because of large numbers of paralogs within protein families. Paralogs that predate the species split (out-paralogs) can easily be confused with true orthologs. Orthologs and in-paralogs are typically detetcted with phylogenetic methods. Alternative approach: ortholog clusters are seeded with a two-way best pairwise match, after which an algorithm for adding in-paralogs is applied.

Software: Inparanoid.

Deepmito: accurate prediction of protein sub-mitochondrial localization using convolutional neural networks (savojardo-2020-deepm)

- Use of deep learning to predict protein localization in four different mitochondrial compartments (matrix, outer, inner and intermembrane regions).

- Dataset = 424 mito. proteins sharing at most 40% sequence identity (CD-HIT filter), including 166 proteins from fungi.

Intertwining phylogenetic trees and networks (schliep-2017-inter)

Bifurcating tree hypothesis (Mindell 2013): the "tree of life" metaphor works well as a strictly bifurcating tree in the absence of reticulate evolution, which results from hybridization, lineage merger, and lateral gene transfer. If this does not hold, phylogenetic networks should be used instead.

Phangorn R package (+ ape): "support value" (nonparametric bootstrap support: Felsenstein 1985; Bayesian posterior probabilities: Rannala & Yang 1996; internode certainty: Salichos, Sta- matakis & Rokas 2014); see also Draper, Hedenäs & Grimm 2007.

An approximately unbiased test of phylogenetic tree selection (shimodaira-2002-approx-unbias)

Related to Shimodaira-Hasegawa test, the AU test adjusts the selection bias overlooked in the standard use of the bootstrap probability and Kishino–Hasegawa tests.

Inparanoid 8: orthology analysis between 273 proteomes, mostly eukaryotic (sonnhammer-2015-inpar)

- 273 species, mainly from Uniprot, including 246 eukaryotes.
- Orthologs may undergo duplications after the speciation event, generating multiple co-orthologs or inparalogs, which complicates orthology detection. InParanoid allows to generate ortholog groups that include all inparalogs but no outparalogs using a parallel 2-pass BLAST procedure.
- quadratic runtime scaling with the number of species.
- Python code for neighbor-joining algorithm.

An exact formula for the number of alignments between two dna sequences (torres-2003-exact-formul)

For two sequences of length 8 and 16 the number of alignments is approx. $40e^6$. See also lange-2002-mathem-statis. General formula:

$$f(n,m) = \sum_{k=0}^{\min(n,m)} 2^k \binom{m}{k} \binom{n}{k}.$$

Note that $\sum_{k=0}^{n} 2^k \binom{n}{k} = 3^n \leq f(n,n) \leq (1+\sqrt{2})^{2n}$.

An l-systems approach to the modelling of fungal growth (tunbridge-1995-l-system)

Complete description of an L-system with rules for ungerminated and germinated spores, tip, apical, septum and ordinary segment, and segment from which a branch grows.

Inferring the mammal tree: species-level sets of phylogenies for questions in ecology, evolution, and conservation (upham-2019-infer)

The authors propose a "backbone-and-patch" approach to tree building applies a newly assembled 31-gene supermatrix to two levels of Bayesian inference: (1) backbone relationships and ages among major lineages, using fossil node or tip dating, and (2) species-level "patch" phylogenies with nonoverlapping in-groups that each correspond to one representa- tive lineage in the backbone.

Scalable Relaxed Clock Phylogenetic Dating (volz-2017-scalab-relax)

Molecular clock models assume a constant substitution rate through time, while relaxed clock models allow robust inference of rates and dates. Such models are usually fitted using Bayesian MCMC, which is computatioanally expensive. This paper explores the relevance of ML and LS phylogenetic and molecular clock dating methods, using a Gamma-Poisson mixture model of substitution rates.

Deep learning for plant genomics and crop improvement (wang-2020-deep)

Advances in statistical modeling: transcriptome-wide association study and deep learning for the prediction of molecular phenotypes from their upstream molecular phenotypes, or directly from genomic DNA sequences.

Convolutional Neural Networks have at least a convolutional layer, which provides them the ability of automatic feature extraction from a contin- uous signal, e.g. DNA/RNA sequence (why is this treated as continuous?!).

See also zou-2019-primer-deep.

A phylogeny-driven genomic encyclopaedia of bacteria and archaea (wu-2009-bacter-archaea)

See also upham-2019-infer and Mike Bostock's Tree of Life.

Molecular phylogenetics: principles and practice (yang-2012-molec)

Molecular phylogenetics is being used to classify metagenomic sequences; to identify genes, regulatory elements, and noncoding RNAs in newly sequenced genomes; to interpret modern and ancient individual genomes; and to reconstruct ancestral genomes. Phylogeny reconstruction methods are either (pairwise) distance- or character-based. Character-based methods include maximum parsimony (MP) — the only approach that is not model-based and thus does not require a substitution model, maximum likelihood (ML) and Bayesian inference (BI) methods. The tree score is the minimum number of changes for MP, the log-likelihood value for ML, and the posterior probability for BI. To reduce complexity, heuristic tree search algorithms are used.

The lack of explicit assumptions, as well as failure to correct for multiple changes at the same site or to accommodate parallel changes on two long branches, renders the parsimony approach somewhat weak. ML has a clear advantage over distance or parsimony methods if we seek to understand the process of sequence evolution. Bayesian phylogenetics makes use of sophisticated models, like the relaxed clock model to infer rooted trees.

Model-based methods (DM, ML and BI) are consistent if the assumed model is correct, while parsimony may be inconsistent under some model-tree combinations.

The bench scientist's guide to statistical analysis of RNA-seq data (yendrek-2012-bench-scien)

Quite outdated; see conesa-2016-survey-best for more up to date material and technologies.

Two methods for mapping and visualizing associated data on phylogeny using ggtree (yu-2018-two-method)

- Two packages: ggtree for mapping and visualization, and treeio for data parsing (Github)
- Bookdown textbook: Data Integration, Manipulation and Visualization of Phylogenetic Trees

See also Letunic I, Bork P. 2007. Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation. Bioinformatics 23:127–128, and iTOL.

A few ascomycota taxa dominate soil fungal communities worldwide (egidi-2020-ascom)

Authors interested in characterizing the identity, global distribution and ecology of dominant soil fungi, based on the analysis of 235 sites distributed across 6 continents. **Results:** 83 dominant fungal phylotypes, where the most ubiquitous fungi found were members of the Pezizomycotina (above 98% match), including Sordariomycetes (*Podospora* and *Chaetomium*), and showed similar level of relative abundance among the sampled habitats.

Clustering genes of common evolutionary history (gori-2016-clust-genes)

Context: Multilocus phylogenetic analysis. Large-scale simulation study of phylogenetic distances and clustering methods to infer loci of common evolutionary history. **Results:** The best-performing combinations are distances accounting for branch lengths followed by spectral clustering or Ward's method.

See also yoshida-2019-multil.

Mapping phylogenetic trees to reveal distinct patterns of evolution (kendall-2016-mappin-phylog)

See also the treespace R package (and maybe apex).

Phyloseq: an r package for reproducible interactive analysis and graphics of microbiome census data (mcmurdie-2013-phylos)

See also mcmurdie-2011-phylos and phyloseq: Explore microbiome profiles using R.

Comparison of phylogenetic trees and search for a central trend in the 'forest of life' (koonin-2011-compar-phylog)

Tree of Life (TOL) vs. "net of life" (because of horizontal gene transfer in prokaryotes). The construction of the trees (about 7000 altogether) provides for an attempt to identify patterns in the Forest of Life (FOL) and address the question whether or not there exists a central trend among the trees that perhaps could be considered an approximation of a TOL. Clustering of the Nearly Universal Trees (NUTs) and the entire Forest of Life (FOL) using Classical MultiDimensional Scaling. **Note:** Boot Split Distance (the contribution of each split is weighted using the bootstrap values)

Treeko: a duplication-aware algorithm for the comparison of phylogenetic tree (marcet-houben-2011-treek)

Two main types of algorithms are available that compute topological distances between phylogenetic trees:

- exploit the topological arrangement of terminal nodes: RF distance, quartet distance, or maximum agreement subtree;
- minimal number of topological re-arrangements: transposition distance, prune and regraft distance, and tree edit distance.

Both approaches require that the mapping of leaves between the trees is complete and univocal. TreeKO is a novel, duplication-aware algorithm that is able to compare two tree topologies re- gardless of the number of duplications and, at the same time, provide a RF-based distance measure that is evolutionarily meaningful and that does not require any prior evolutionary assumption. In addition treeKO, implements the possibility of computing so-called phylome support values (22), and reconciliation-based measures such as the number of inferred duplications and losses events.

See also ete-compare.

Ete 3: reconstruction, analysis, and visualization of phylogenomic data (huerta-cepas-2016-ete)

Website: `http://etetoolkit.org` Basic recipe to build a custom workflow: `aligner-trimmer-tester-builder`.

MUSCLE: multiple sequence alignment with high accuracy and high throughput (edgar-2004-muscl)

MUSCLE relies on 2 distance measures (k-mer and Kimura distance), and distance matrices are clustered using UPGMA (and not NJ, because "by assuming that in progressive alignment, the best accuracy is obtained at each node by aligning the two profiles that have fewest differences, even if they are not evolutionary neighbors."). The scoring function defined on aligned pairs of profile positions is the log-expectation (LE) score, which is a modified version of the log-average function. Position-specific gap penalties are used, similar to MAFFT.

See other tools for Multiple Sequence Alignment.

Towards a reliable objective function for multiple sequence alignments (thompson-2001-towar-reliab)

The authors propose a new objective scoring function for multiple sequence alignments (norMD), which combines the advantages of the column- scoring techniques with the sensitivity of methods incorporating residue similarity scores. Compared to other approaches, norMD incorporates ab initio sequence information, such as the number, length and similarity of the sequences to be aligned.

Physiological significance of network organization in fungi (simonin-2012-physiol-signif)

Hyphal fusion is necessary for efficient nutrient translocation in Neurospora crassa. Techniques used: radioisotope trac- ers, stable isotope tracers, and fluorescently labeled proteins, to visualize and quantify nutrient flows and cytoplasmic and nuclear movement. Analysis considers strips at varying excentricity (1.67-cm regions).

Combining multiple tools outperforms individual methods in gene set enrichment analyses (alhamdoosh-2017-combin)

Description of a new approach: ensemble of genes set enrichment analyses (EGSEA), which uses 12 algorithms and calculates collective gene set scores. Also provides a brief review of existing approaches: ORA methods vs. enrichment score-based methods or permutation-based testing (including variant thereof, e.g. the "camera method").

See khatri-2012-ten-years regarding limitations of the ORA approach.

A comparison of methods for differential expression analysis of rna-seq data (soneson-2013-rna)

This article provides a comparison of eleven methods for differential expression analysis of RNA-seq data.

- **DESeq** is conservative with default settings. It becomes more conservative when outliers are introduced. It generally has low TPR and poor FDR control with 2 samples/condition, while maintaining good FDR control for larger sample sizes, also with outliers. It has medium computational time requirement, increases slightly with sample size.
- **edgeR** has slightly liberal for small sample sizes with default settings. It becomes more liberal when outliers are introduced. It generally has high TPR and poor FDR control in many cases,

worse with outliers. Medium computational time requirement, largely independent of sample size.

See anders-2013-count-rna for a detailed description of the above R packages.

Local pore size correlations determine flow distributions in porous media (alim-2017-local-pore)

The idea of this paper is that local correlations between adjacent pores determine the distribution of fractions of fluid flow propagated from one pore to others that are downstream, and when there is sufficient disorder at the pore scale, these flow rates are partitioned randomly. A combination of two singular and a continuous distributions in flow fractions gives rise to an exponential decay generalizing results previously obtained in a similar description of force distributions in random bead packs.

Mechanism of signal propagation in physarum polycephalum (alim-2017-mechan-physar)

The authors discuss a mathematical model whereby a front of increased contraction propagates with a velocity comparable to the flow-driven dispersion of particles. This model is well suited to explain the adaptation of the peristaltic wave to organism size and *P. polycephalum* ability to find the shortest route between food sources. Like in alim-2017-local-pore it relies on the hypothesis of mass conservation.

> Key time-variant parameters were calculated from every point of a network using custom MATLAB (The MathWorks) code. Briefly, an image from a time series was thresholded to extract just the plasmodial tubes then skeletonized and separated into a connected map of nodes and edges. The skeleton was used to recover the tube radius at every point by finding the largest circle that would still entirely fit within a tube.

Reading notes

A common-sense guide to data structures and algorithms (wengrow-2017-common-sense)

Short but interesting textbook on general programming principles. It features a mix of Python and JavaScript to illustrate major sorting algorithms, big O notation, etc. Available as EPUB locally.

A Flexible Parametric Accelerated Failure Time Model (crowther-2020-flexib-param)

Accelerated Failure Time (AFT) models as an alternative to classical PH models. Under the AFT model, the effect of covariates act to accelerate or decelerate the time to event of interest. Time-dependent acceleration factors, as well as time-dependent covariates (e.g. delayed entry) are also allowed. Stata and R packages available, as well as Python code.

A statistical analysis of probabilistic counting algorithms (clifford-2010-statis-analy)

See also ertl-2017-new-hyper and SlowerLogLog by Evan Miller.

Agile Data Science (jurney-2014-agile-data-scien)

Keywords: scalability, NoSQL (Hadoop and MongoDB), cloud computing, big data, data intuition Interesting use of personal email data:

> In Agile Big Data, a small team of generalists uses scalable, high-level tools and cloud computing to iteratively refine data into increasingly higher states of value. We embrace a software stack leveraging cloud computing, distributed systems, and platforms as a service. Then we use this stack to iteratively publish the intermediate results of even our most in-depth research to snowball value from simple records to predictions and actions that create value and let us capture some of it to turn data into dollars.

See also A manifesto for Agile data science.

Sidenote: There is an example of using the Enron SQL database (Chapter 2, § "SQL").

Algorithmic Mathematics (soicher-2004-algor-mathem)

Basic overview of number theory and related algorithms, with several exercises and tips at the end.

Algorithms (erickson-2018-algor)

See also:

- Margaret M. Fleck. Building Blocks for Theoretical Computer Science. Version 1.3 (January 2013)
- Eric Lehman, F. Thomson Leighton, and Albert R. Meyer. Mathematics for Computer Science. June 2018 revision

- Pat Morin. Open Data Structures. Edition 0.1Gb (January 2016)
- Don Sheehy. A Course in Data Structures and Object-Oriented Design. February 2019 or later revision

Russian (Peasant) multiplication (See also Egyptian Multiplication)

```
def peasant(x, y):
    z = 0
    while y > 0:
        if y % 2 == 1:
            z += x
        x <<= 1
        y >>= 1
    return z
```

Also know as **Ethiopian multiplication**, see, e.g. Rosetta:

```
halve  = lambda x: x // 2
double = lambda x: x * 2
even   = lambda x: not x % 2

def ethiopian(m, n):
    result = 0
    while m >= 1:
        if not even(m):
            result += n
        m = halve(m)
        n = double(n)
    return result
```

Quick translation in Scheme (FIXME):

```
(define-syntax (while stx)
  (syntax-case stx ()
      ((_ condition expression ...)
       #'(do ()
           ((not condition))
           expression
           ...))))

(define (peasant x y)
  (let ((z 0))
  (while (> y 0)
    (if (odd? y) (set! z (+ z x)))
    (bitwise-arithmetic-shift-left x 1)
```

```
    (bitwise-arithmetic-shift-right y 1))
  z))
```

Algorithms Unlocked (cormen-2013-algor-unloc)

> We want two things from a computer algorithm: given an input to a problem, it should always produce a correct solution to the problem, and it should use com- putational resources efficiently while doing so.

- exact vs. approximate solution (e.g., RSA and large prime numbers)
- focusing on the order of growth of the running time as a function of the input size
- algorithms described in plain English, and not in pseudo-code like in CLRS

An incremental approach to compiler construction (ghuloum-2006-increm-approac)

Found by following Thorsten Ball's progress (on Twitter) on his approach to build a Scheme compiler from scratch.

An introduction to bioinformatics algorithms (jones-2004-introd-bioin-algor)

The authors make use of simplified pseudo-code for all the algorithms discussed in this book – on the basis that the target audience are biologists. I found it nice, as it is heavily inspired from Python syntax (significant indentation is fine for reading purpose, IMHO). The introductory chapter on computer science (CS) is pretty basic stuff that can be found in any introductory textbook (chapter 2): algorithmic complexity, recursive versus iterative approach, type of algorithms (brute force, branch-and-bound, greedy approach, dynamic programming, divide-and-conquer, machine learning, randomized algorithms), and NP-completeness. It is intended for biologists.

> I have indeed been able to apply my skills in this new area, but only after coming to understand that solving biological problems requires far more than clever algorithms: it involves a creative partnership between biologists and mathematical scientists to arrive at an appropriate mathematical model, the acquisition and use of diverse sources of data, and statistical methods to show that the biological patterns and regularities that we discover could not be due to chance. — Richard Karp

For CS folks, the third chapter provides a gentle primer to biology.

See also Bioinformatics Algorithms, by Saad Mneimneh, which offers solutions to selected exercises from each chapter.

Analytic combinatorics for bioinformatics I: seeding methods (filion-2017-analy-combin)

See also Calibrating seed-based alignment heuristics with Sesame, and Choosing the best heuristic for seeded alignment of DNA sequences.

Applied Data Science (langmore-2012-applied-data-scien)

Nice applied textbook on "data science" using Unix tools and Python. This is the first time I saw linear regression introduced using Bayesian formalism, then regularization. Lasso penalization is discussed in the case of LOgistic regression. There's also an interesting chapter on high-performance Python (p. 106 ff.).

See also Data science: An action plan for expanding the technical areas of the field of statistics, by Cleveland:

- **Multidisciplinary Investigations** (25%): data analysis collaborations in a collection of subject matter areas.
- **Models and Methods for Data** (20%): statistical models; methods of model building; and methods of estimation and distribution based on probabilistic inference.
- **Computing with Data** (15%): hardware systems; software systems; and computational algorithms.
- **Pedagogy** (15%): curriculum planning and approaches to teaching for elementary school, secondary school, college, graduate school, continuing education, and corporate training.
- **Tool Evaluation** (5%): surveys of tools in use in practice, surveys of perceived needs for new tools, and studies of the processes for developing new tools.
- **Theory** (20%): foundations of data science; general approaches to models and methods, to computing with data, to teaching, and to tool evaluation; mathematical investigations of models and methods, of computing with data, of teaching, and of evaluation.

Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition (dorie-2018-autom)

Focus on semi-parametric and nonparametric causal inference methodology, with a particular emphasis on the comparison between

30 different approaches through the "causal inference data analysis competition", hosted during the 2016 Atlantic Causal Inference Conference Competition.

Some caveats when assessing causal inference methods: (1) few methods compared and unfair comparisons, (2) testing grounds not calibrated to "real life", and (3) file drawer effect. The later ressembles what is commonly impacting meta-analytical studies. It reminds me of a critic of machine elarning algorithms that are always developed and calibrated on exiting data sets, like those available on UCI, with reference to existing benchmarks—hence inducing a confirmation bias—and that would probably perform poorly on real life data (I didn't find the reference). See also this online article, Controlling machine-learning algorithms and their biases, by Tobias Baer and Vishnu Kamalnath, regarding human biases.

See also: middleton-2016-bias-amplif.

Sidenote: Omitted variable bias

Suppose the true model is $Y = \alpha_0 + \alpha_1 X + \alpha_2 Z + u$, and we estimate $Y = \beta_0 + \beta_1 X + u$. Then the omitted variable can be considered as a function of X in a conditional regression $Z = \gamma_0 + \gamma_1 X + w$. So we have estimated

$$

$$\begin{aligned} Y &= \beta_0 + \beta_1 X + \beta_2(\gamma_0 + \gamma_1 X + w) + u \\ &= (\beta_0 + \beta_2\gamma_0) + (\beta_1 + \gamma_1\beta_2)X + (\beta_2 w + u) \end{aligned}$$

$$

Unless $\beta_2 = 0$, $\mathbb{E}(\hat{\beta}_1) = \beta_1 + \beta_2 \left(\frac{\sum xz}{\sum x^2}\right) \neq 0$, which means that the coefficient of X picks up the part of the influence of Z that was correlated with X.

Bigtable: a distributed storage system for structured data (chang-2006-bigtab)

> Bigtable does not support a full relational data model; instead, it provides clients with a simple data model that supports dynamic control over data layout and format, and allows clients to reason about the locality properties of the data represented in the underlying storage. Data is indexed using row and column names that can be arbitrary strings. Bigtable also treats data as uninterpreted strings, although clients often serialize various forms of structured and semi-structured data into these strings. Clients can control the locality of their data through careful choices in their schemas. Finally, Bigtable schema parameters let clients dynamically control whether to serve data out of memory or from disk.

Bioinformatics data skills: reproducible and robust research with open source tools (buffalo-2015-bioin-data-skill)

- Sequence Read Archive
- forensic bioinformatics (Baggerly and Coombes 2009)

Bootstrap Confidence Intervals (diciccio-1996-boots-confid-inter)

Four bootstrap confidence interval procedures: BCa, bootstrap-t, ABC and calibration. See the bootstrap R package for ABC and `boot::abc.ci` for calibrated ABC.

Bootstrap Confidence Levels For Phylogenetic Trees (efron-1996-boots-confid)

One of the many applied papers on the bootstrap by Efron, based on the original work of Felsenstein (see also felsenstein-2004-infer-phylog). The aim of bootstrap resampling in phylogenetic reconstruction is to assess the confidence for each clad, based on the proportion of bootstrap trees showing that same clade. In this context, the notion of agreement refers to the topology of the trees and not to the length of its arms. The rationale underlying the bootstrap confidence values depends on a simple multinomial probability model, although a bivariate normal model could also be used (parametric bootstrap).

Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy (efron-1986-boots-method)

From the Stata Manual [R] on "bootstrap": efron-1986-boots-method describe an alternative to Satterthwaite's approximation that estimates the ASL by bootstrapping the statistic from the test of equal means. Their idea is to recenter the two samples to the combined sample mean so that the data now conform to the null hypothesis but that the variances within the samples remain unchanged.

```
summarize mpg, meanonly
scalar omean = r(mean)
summarize mpg if foreign==0, meanonly
replace mpg = mpg - r(mean) + scalar(omean) if foreign==0
summarize mpg if foreign==1, meanonly
replace mpg = mpg - r(mean) + scalar(omean) if foreign==1
by foreign, sort: summarize mpg
keep mpg foreign
set seed 1
bootstrap t=r(t), rep(1000) strata(foreign) saving(bsauto2) nodots: ttest mpg, by(foreign) unequal
```

See also hesterberg-2014-what-teach and Patrick Burns note on resampling. See also poi-2004-from-help-desk and the corresponding entry for R code.

Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation (tsamardinos-2017-boots-out)

Bootstrap Bias Corrected CV = bootstrap the whole process of selecting the best-performing configuration on the out-of-sample predictions of each configuration, without additional training of models. Computationally more efficient, smaller variance and bias compared to nested CV.

Clojure Data Analysis Cookbook (rochester-2013-clojur-data)

A book from the Packt Publishing group.

Actually, this is the first book by Eric Rochester. The second covers more advanced techniques and was published one year later: cite:rochester-2014-master-clojur. The site for the book includes data used throughout the book, nothing more, but be aware there are a lot of datasets.

> This book is for programmers or data scientists who are familiar with Clojure and want to use it in their data analysis processes.

The first chapter describes various ways to import data (flat files, local database and RDF data), mostly using Incanter backend. I would prefer the author start with more basic tool before dwelling into specialized libraries, especially since Incanter looks almost defunct nowadays (the last blog entry I found said that it was version 1.5.7, Feb 2016). Anyway, this provides a good overview of Incanter's facilities to process external data and convert them in array form, and R or Lispstat users should feel at home. However, starting with Chapter 2 the author will use the data.csv library.

Clojure for the Brave and True (higginbotham-2015-clojur-brave-true)

The book was published on Leanpub a while ago but it is not for sale anymore. I don't remember where I got a PDF version of the book, but there is also a website, Brave Clojure, where the book can be read online for free.

The first chapters are all about setting up a working environment for writing Clojure code, and it happens to be Emacs + Cider. The Clojure version currently used in the book is 1.6 (alpha3), with Leiningen as the build tool for Clojure projects (+ Clojure 1.5.1 for `lein repl`).

Overall, the presentation is clear although it remains a bit rough (I mean like in draft mode) with lot of external links to learn more.

Competitive programmer's handbook (laaksonen-2017-compet-progr-handb]])

When I first came across this textbook, the title reminded me of The Imposter Handbook. Unlike conery-2016-impos-hand, it has more running code, and in a decent language (C++ 11). I wrote a little transcript in Python 3.x and wrote a review on aliquote.org.

Concrete abstractions: an introduction to computer science using scheme (hailperin-1999-concr-abstr)

TODO Post a review on `http://aliquote.org`.

Dancing Links (knuth-2000-dancin-links)

`https://dancing-links.herokuapp.com`

Data Wrangling with Python (kazil-2016-data-wrang-python)

Relatively self-paced introduction to Python data structures and programming. In order to motivate the reader, the authors said they would understand the following three lines by the end of chapter 2, and I believe this should be true even for people who know close to nothing to programming.

```
import sys
import pprint
pprint.pprint(sys.path)
```

> You just learned how to program. Programming is not about memorizing everything; rather, it is about troubleshooting when things go awry.

Designing Data-Intensive Applications (kleppmann-2016-desig-data)

Review by Henrik Warne.

From the help desk: some bootstrapping techniques (poi-2004-from-help-desk)

Hypothesis test based on bootstrap resampling:

```
x1 <- d[,1] - mean(d[,1]) + mean(x)
x2 <- d[,2] - mean(d[,2]) + mean(x)
B <- 10000        ## no. bootstrap samples
```

```
s <- numeric(B)   ## vector of test statistics
for (i in 1:B) {
  x1s <- sample(x1, replace=TRUE)
  x2s <- sample(x2, replace=TRUE)
  s[i] <- mean(x1s) - mean(x2s)
}
pobs <-  (1 + sum(abs(s) > abs(s0))) / (B+1)
```

See also efron-1986-boots-method.

Graph-based genome alignment and genotyping with hisat2 and hisat-genotype (kim-2019-graph-hisat)

HISAT2 is the successor of TopHat2. What's new? HISAT2 can align both DNA and RNA sequences using a graph Ferragina Manzini index. This graph-based alignment approach enables much higher alignment sensitivity and accuracy than standard, linear reference-based alignment approaches, especially for highly polymorphic genomic regions.

Haskell Programming from First Principles (allen-2016-haskel-progr)

One of the best book I read about Haskell so far, and on functional programming more generally.

A short remark about typography: this book is typesetted using LaTeX; however, the verbatim and math elements appear a bit too small in my view.

How many imputations do you need? A two-stage calculation using a quadratic rule (hippel-2016-how)

See also `https://statisticalhorizons.com/how-many-imputations`.

1. First, carry out a pilot analysis. Impute the data using a convenient number of imputations. (20 imputations is a reasonable default, if it doesn't take too long.) Estimate the FMI by analyzing the imputed data.
2. Next, plug the estimated FMI into the formula above to figure out how many imputations you need to achieve a certain value of CV(SE). If you need more imputations than you had in the pilot, then add those imputations and analyze the data again.

Ideal Hash Trees (bagwell-2001-ideal-hash-trees)

See also Hash Array Mapped Tries and Bodil Stokke's talk, Meeting with Remarkable Trees.

Immutability Changes Everything (helland-2015-immut-chang-every)

Append-only computing: The truth is the log. The database is a cache of a subset of the log.

Implementation strategies for continuations (clinger-1988-implem-strat-contin)

Continuations (`call/cc` in Scheme) are generally used to manipulate the flow of control in a program, which means that they are close to `GOTO` statements in imperative languages. See 3.3. Continuations of TSPL3 and 10.4 Continuations of the Racket Reference guide.

Improving Palliative Care with Deep Learning (avati-2017-improv-palliat)

See Frank Harrell's blog post.

> As with any retrospective study not based on an inception cohort with a well-defined "time zero", it is tricky to define a time zero and somewhat easy to have survival bias and other sampling biases sneak into the analysis. The ML algorithm required division of patients into "positive" and "negative" cases, something not required by regression models. "Positive" cases must have at least 12 months of previous data in the health system, weeding out patients who died quickly. "Negative" cases must have been alive for at least 12 months from the prediction date. It is also not clear how variable censoring times were handled. In standard statistical model, patients entering the system just before the data analysis have short follow-up and are right-censored early, but still contribute some information.

Living Clojure (meier-2015-livin-clojur)

See also How I start.

Loving Common Lisp (watson-2016-lovin-common-lisp)

On Github (depends on clml), cloned locally in `file:///Users/chl/git/sandbox`.

There are still some proof-reading lacking here and there but overall it is quite readable. The very first part of the book is all about data types in Common Lisp. All examples are illustrated using SBCL.

The author does not explain the differences between defvar, defparameter, setf and setq, although they are used a lot interchangeably at the beginning of the book. Treatment of lists is pretty standard (`car` and `cdr`, `cons` and `append`, `last` and `nth`, etc.). An interesting example regarding shared structure in list is provided:

```
(setq x '(0 0 0 0))
(setq y (list x x x x))
(setf (nth 2 (nth 1 y)) 'x)
x
y
(setq z '((0 0 0 0) (0 0 0 0) (0 0 0 0)))
(setf z (nth 2 (nth 1 z)) 'x)
z
```

Beyond lists, vectors and arrays (`make-array`, or `vector` and `make-sequence`) are more efficient data structure when the number of elements is large. Beware that CL for scientific computing cannot be fast, portable, and convenient all at the same time. Notice that an array can "contain" any values, and thus mixing integers with float is allowed by the language.

```
(defvar y (make-array '(2 3) :initial-element 1))
(setf (aref y 1 2) 3.14159)
y
```

Operations on string (`concatenate`, `search`, `subseq` and `string-*`) and the fine distinction between `eq`, `eql`, and `equal`. See also Lisp - List Processing (or Lots of Irritating Superfluous Parenthesis). For strings, we should prefer `string=`. Instead of `nth`, we use `char` to extract a given character in a string.

Hash tables are to be preferred when lists (coupled with `assoc`) are long. Main functions are `gethash`, `make-hash-table`, and `maphash`. Updating values in a hash table is done using `remhash` or `clrhash`. Note that these functions can modify their arguments, much like `setf` or `setq`, but the latter are macros and not functions.

> Functional programming means that we avoid maintaining state inside of functions and treat data as immutable.

Recall that read-only objects are inherently thread safe.

Lisp functions: `defun`, keywords (`&aux`, `&optional`, `&key`), `let` special operator for local bindings, `lambda` and `funcall`.

```
(defvar f1 #'(lambda (x) (+ x 1)))
(funcall f1 100)
```

A closure is a function that references an outer lexically scoped variable, which typically happens when functions are defined inside `let` forms (see p. 47).

The `dotimes` and `dolist` macros are close to Stata `forvalues` and `foreach` instructions. The `do` macro is more general:

```
(do ((i 0 (1+ i)))
    ((> i 3) "value-of-do-loop")
  (print i))
```

Input (`*standard-input*`) and output (`*standard-output*`) of Lisp data is handled using streams, and the `with-open-file` macro. Note that it is possible to use `make-pathname` to build a proper absolute or relative path, instead of using (quoted) strings. Here is a typical example of reading a file line by line:

```
(defun readline ()
  "Read a maximum of 1000 expressions from the file 'test.dat'"
  (with-open-file
    (input-stream "test.dat" :direction :input)
    (dotimes (i 1000)
      (let ((x (read-line input-stream nil nil)))
        (if (null x) (return))
        (format t "next line in file: ~S~%" x)))))
```

The rest of the book describes some application of web and network programming using CLOS classes and various packages (`drakma`, `hunchentoot`). The chapter of querying database is also interesting.

Machine learning in python: main developments and technology trends in data science, machine learning, and artificial intelligence (raschka-2020-machin-learn-python)

Interesting review of current data stack in Python. The first part focus on scikit-learn and contrib, "classical ML" approaches, including boosting machines (LightGBM), and distributed computing using Dask-ML. Little is said about H2O and the Sparkling Water Spark-adapter, though. AutoML libraries include: Auto-Weka, Auto-sklearn, TPOT, H2o-AutoML, AutoKeras.

See also he-2020-autom.

Mature Optimization Handbook (bueno-2013-matur-optim)

Review published on aliquote.org.

Models in biology: 'accurate descriptions of our pathetic thinking' (gunawardena-2014-model)

Emphasizes the role of forward modeling, especially with respect to causality.

> Mathematical models come in a variety of flavors, depending on whether the state of a system is measured in discrete units ('off' and 'on'), in continuous concentrations or as probability distributions and whether time and space are themselves treated discretely or continuously.

Modern Vim: Craft Your Development Environment with Vim 8 and Neovim (neil-2018-moder-vim)

This is mostly about Neovim, but there are many references to Vim; sort of, what's available in Neovim that has been incorporated in Vim, except for package management. The author describes the builtin plugin system (no need for pathogen or vim-plug), the FZF plugin (instead of Ctrl-P) — I have no interest in semantic organization of files in a project (`tpope/vim-projectionist`), how to use the quickfix list (with tmux adapter), and the builtin terminal emulator (there's no insert mode, instead it's called "terminal mode"; use `C-\ C-n` to toggle between Terminal and Normal mode). I didn't read the chpater about sessions because I don't need them, but overall I like this book a lot: the style is clear and concise and the examples are well put. The appendix provides interesting discussion regarding Language Server Protocol in Vim. There's also a brief discussion on the future of Vim 8 (and Neovim).

> Vim is not a shell or an Operating System. You will not be able to run a shell inside Vim or use it to control a debugger. This should work the other way around: Use Vim as a component from a shell or in an IDE.
> — Bram Moolenaar (Vim documentation)

Sidenote:
Useful packages and config for Lisp editing:

- `https://mendo.zone/fun/neovim-setup-haskell/`
- `https://github.com/Shougo/deoplete.nvim`
- `https://github.com/kovisoft/slimv`
- `https://blog.venanti.us/clojure-vim/`

New cardinality estimation algorithms for hyperloglog sketches (ertl-2017-new-hyper)

See also SlowerLogLog by Evan Miller.

Novel parallel algorithm for constructing Delaunay triangulation based on a twofold-divide-and-conquer scheme (wu-2014-novel-delaun)

Multitasking parallel algorithm, in 3 stages: This algorithm automatically divides the planar point set into several non-overlapping

subsets along the x-axis and y-axis directions alternately, according to the number of points and their spatial distribution. Next, the Guibas–Stolfi divide-and-conquer algorithm is applied to construct Delaunay sub- triangulations in each subset. Finally, the sub-triangulations are merged based on the binary tree.

See also:

- The Delaunay's Dual and d3-delaunay
- Lloyd's Algorithm
- Delaunay Triangulation
- Voronoi Topology
- Implementation of Voronoi Diagram and Delaunay Triangulation

Numerical issues in statistical computing for the social scientist (altman-2004-numer-issues)

Although it is probably a bit outdated by now, I like to refer to this book when it comes to summarize how important dedicated statistical packages are compared to, say, MS Excel (which used a single-pass formula for computing the SD of a series of values). More to the point, statistical software dedicated to survey analysis provide better estimates than more general package, except perhaps Stata which has good estimators of variance for complex surveys.

Sources of inaccuracy in statistical computation: bugs, computer arithmetic, randomized algorithms, approximation and heuristic algorithms, local search algorithms. About computer arithmetic, specifically:

> There's a credibility gap: We don't know how much of the computer's answers to believe. Novice computer users solve this problem by implicitly trusting in the computer as an infallible authority; they tend to believe that all digits of a printed answer are significant. Disillusioned computer users have just the opposite approach; they are constantly afraid that their answers are almost meaningless. — Don Knuth

Take away message from computer arithmetic:

1. Rounding errors occur in binary computer arithmetic that are not obvious when one considers only ordinary decimal arithmetic.
2. Round-off error tends to accumulate when adding large and small numbers — small numbers tend to "drop off the end" of the addition operator's precision, and what accumulates in the leftmost decimal positions is inaccurate.

3. Subtracting a similar quantity from the result can then "cancel" the relatively accurate numbers in the rightmost decimal places, leaving only the least accurate portions.

   Illustration: $i = 1000000000 + 2 - 0.1 - 1000000000$.

   Side note: The failure of SAS to recover true coefficients of a rare count event model in Table 1.2 should be checked with more recent version of SAS.

Open problems in algebraic statistics (sturmfels-2007-open-probl)

Open problems at the intersection between interactions between algebraic geometry and computational statistics. E.g., Graphical Models with Hidden Variables: Our first question concerns three-dimensional contingency tables (p_{ijk}) whose indices i, j, k range over a set of four elements, such as the set A, C, G, T of DNA bases. Consider the variety of 4×4×4-tables of tensor rank at most 4. There are certain known polynomials of degree at most nine which vanish on this variety. Do they suffice to cut out the variety?

See also: pistone-2001-algeb-statis, gibilisco-2010-algeb-geomet.

Orthology detection combining clustering and synteny for very large datasets (lechner-2014-orthol-detec)

- orthology is not a transitive relation so that the problem is different from clustering an input gene set.
- the authors focus on avoiding false positive orthology assignments within the phylogenetic range of the reported orthologous groups, while tolerating recent in-paralogs (speciation preceding duplication) as unavoidable contamination

Overly optimistic prediction results on imbalanced data: flaws and benefits of applying over-sampling (vandewiele-2020-overl-optim)

Methodological bias = applying over-sampling before partitioning the data into mutually exclusive training and testing sets. Other biased approaches: apply cross-validation on a subset of data subsampled from the original dataset (increases the variance of the obtained results and does not address class imbalance). Carrying out over-sampling before splitting into training and testing sets might leak information from the original testing samples to the artificially generated training samples, leading to overly optimistic validation scores. It is therefore of key importance to carry out the over-sampling after selecting a training and testing set.

Parallel computing with r: a brief review (eddelbuettel-2019-paral-comput)

Standard HPC stilla round, but it is nowadays overshadowed by cloud computing; Haddop, Spark; deep learning. Bengtsson's `future` package offers a nice abstraction to local and remote parallelism options. A key aspect of concurrency is the *task-switching cost*. Single instruction multiple data (SIMD) and the AVX-512 instruction sets are another example of CPU- and compiler-centric parallel instructions. OpenMP remains a key technology for parallel execution of compiled code. Note that parallel execution requires stream-aware RNGs (p.7).

Power-law distribution in empirical data (clauset-2009-power)

1. Estimate the parameters xmin and α of the power-law model using the methods described in Section 3.
2. Calculate the goodness-of-fit between the data and the power law using the method described in Section 4. If the resulting p-value is greater than 0.1 the power law is a plausible hypothesis for the data, otherwise it is rejected.
3. Compare the power law with alternative hypotheses via a likelihood ratio test, as described in Section 5. For each alternative, if the calculated likelihood ratio is significantly different from zero, then its sign indicates whether the alternative is favored over the power-law model or not.

Automl-zero: evolving machine learning algorithms from scratch (real-2020-autom-zero)

Github: `https://github.com/google-research/google-research/tree/master/automl_zero`

See also he-2020-autom.

Prediction, estimation, and attribution (efron-2020-predic-estim-attrib)

This article deals with the controversy around prediction versus explanation in statistics and machine learning communities, as a sequel of breiman-2001-statis-model. How do the pure prediction algorithms relate to traditional regression methods? In traditional approaches to regression modeling a description of the underlying scientific truth (the "surface") is formulated, along with a model of the errors that obscure direct observation ("surface plus noise formulation"). On the contrary pure prediction algorithms focus on prediction, to the neglect of estimation and attribution. The idea of

boosting is, for example, to have many weak learners that combine effectively to yield low error rate while traditional methods focus on strong individual predictors.

Interesting note on GWAS analysis: Instead of performing a traditional attribution analysis with p = 106 predictors, the GWAS procedure performed 106 analyses with p = 1 and then used a second layer of inference to interpret the results of the first layer. See also comment on local false discovery rate.

Random forests, decision trees, and categorical predictors: the "absent levels" problem (au-2018-random-fores)

This paper discusses the case of how best to handle catgeorical predictors in RF, in particular the 'absent level' problem, i.e. the case of the indeterminacy over how to handle an observation that has reached a categorical split which was determined when the observation in question's level was absent during training.

Reaching python from racket (ramos-2014-reach-python-racket)

Via Racket News #15. See also Racket is an acceptable Python.

Reconciling modern machine learning practice and the bias-variance trade-off (belkin-2019-recon)

Interesting article on the bias-variance tradeoff in the context of recent ML workflows (NNs, deep learning, etc.). The authors discussed the "unified performance curve" and present compelling evidence that increasing model capacity beyond the point of interpolation results in improved performance in several use cases.

Maybe see murphy-2012-machin-learn.

Representing numeric data in 32 bits while preserving 64-bit precision (neal-2015-repres-numer)

Every number with up to seven significant decimal digits maps to a distinct 32-bit single precision value, with no information loss. However, when these single precision values are converted to 64-bit double precision in the standard (hardware-supported) way and then used in arithmetic operations, the results are in general not the same as if a 64-bit floating-point representation had been used. The problem is that the standard conversion by extending the mantissa of a single precision number with zeros does not produce the correct double precision representation of a number, such as 0.1, whose binary expansion is non-terminating. As an alternative

we might consider using decimal floating point but floating point division operation required to convert from a decimal floating point representation is quite slow.

Cowlishaw, M. F. (2003) "Decimal Floating-Point: Algorism for Computers", in Proceedings of the 16th IEEE Symposium on Computer Arithmetic.

Second thoughts on the bootstrap (efron-2003-secon-thoug-boots)

Plug-in principle: travel from the real world to the bootstrap world simply by plugging in a point estimate $\hat{P}$ for P. This is the only inference step. There may be a problem with the miscentering of the $\hat{\sigma}^*$ values, as exemplified by the "dilatation phenomenon", like with Stein's estimation. Second-order (bootstrap t and BCA) accuracy suggested that the bootstrap could provide good approximate confidence intervals, better than the standard $\hat{\theta} \pm z_\alpha \hat{\sigma}$.

> Personally my biggest bootstrap surprise involved the ABC intervals developed with Tom DiCiccio in 1992. The ABC is an analytic approximation to the BCA method that was intended to cut down on the 2000 or so bootstrap simulations required for BCA. In fact, ABC involves no simulation at all, which was the surprise, especially since the method gives excellent results for smoothly differentiable statistics like the correlation coefficient.

Illustration with phylogenetic trees: "the tree is a statistic, admittedly a complicated one, and it is reasonable to ask how much trust we can place in the observed features." The statistical interpretation of Felsenstein's confidence values (whereby the columns of x are resampled, bootstrap trees are constructed and the proportion of bootstrap trees that have the feature of interest simply are counted) is discussed in efron-1996-boots-confid.

Serious Python (danjou-2018-serious-python)

Nice book to understand the underside of Python, especially regarding package import and path management. Note that this will not teach you Python programming, but it will certainly be helpful to better understand Python, think about design patterns, and how to develop your own projects. Each chapter provides a discussion of important topics in project development, and a brief interview by core developers is provided at the end. Note that some chapters are very specific of some aspects of Python programming, or PL more generally. For instance, chapter 4 deals with timestamp and the importance of timezone.

I learned a few things about packaging, and in particular the number of modules that were developed before `pip`, namely (in chrono-

logical order): `distutils`, `setuptools`, `distribute`, `distutils2`, `packaging`, and `distlib`. The latter may eventually replace `setuptools`.

Sick individuals and sick populations (rose-2001-sick)

A good question to ask is "Why did this patient get this disease at this time?", since it also implies that we care about why it happened and whether it could have been prevented. The individual-centered approach leads to the use of RR, but this approach to the search of causes has to assume heterogeneity of exposure within the study population.

> If everyone smoked 20 cigarettes a day, then clinical, case-control and cohort studies alike would lead us to conclude that lung cancer was a genetic disease; and in one sense that would be true, since if everyone is exposed to the necessary agent, then the distribution of cases is wholly determined by individual susceptibility.

Sparse data bias: a problem hiding in plain sight (greenland-2016-spars-data-bias)

When the data lack adequate case numbers for some combination of risk factor and outcome levels, the resulting estimates of the regression coefficients can have bias away from the null, hence the term "sparse data bias" because it is not limited to small samples.

Causes:

- Few outcome events per variable (EPV), as measured by the number of failures per variable for Cox proportional hazards and Poisson regression, and the minimum of the numbers of cases and non-cases per variable for logistic regression (for conditional logistic regression, only the numbers within discordant matched sets should be counted)
- Variables with narrow distributions or with categories that are very uncommon
- Variables that together almost perfectly predict the outcome (eg, if a combination of discrete covariate levels is found only among the study participants with outcome)
- Variables that together almost perfectly predict the exposure (eg, if a combination of discrete covariate levels is found only among the study participants who are exposed).

Solutions:

- Stepwise variable selection procedures

- Exact statistical methods (eg, exact logistic regression)
- Exposure or treatment modelling (eg, propensity scoring, inverse-probability-of- treatment weighting)
- Penalisation

Penalization produces the most accurate estimates given the information in the penalty; data augmentation version is simple and feasible in all statistical software; can be used as a diagnostic tool for sparse data bias.

Statistical computing and databases: distributed computing near the data (chen-2003-statis-comput-datab)

Old stuff but interesting ideas (part of them are now materialized in the dplyr/dbi packages) like performing the data-intensive but algorithmically less sophisticated operations in the database and send back the results to the statistical package which is responsible for the algorithmic flow. The software design includes a CORBA architecture coupled to PVM for managing parallel computations.

Statistical methods need software: a view of statistical computing (ripley-2002-statis-method)

> Let's not kid ourselves: the most widely used piece of software for statistics is Excel.

Statistical Software Certification (gould-2001-statis-softw-certif)

> Stata is instead tested using an automated procedure that involves running 1,064 do-files containing 158,391 lines that cause Stata to execute 38,343,139 commands and produces just over 16 megabytes (473,859 lines) of output.

Mostly about the internal process of certification *per se* rather than scientific computing, except maybe p. 40 ff when the author discuss the problem of false precision: Double precision floating point numbers are stored using 64 bits. Coprocessors, however, use 80 bits, providing extra guard bits to improve accuracy. On the coprocessor, calculations are made using 80 bits and are then handed back to the CPU rounded to 64 bits.

According to altman-2004-numer-issues, Stata is quite good. For instance, Stata v6 correctly returned the certified values for the Pi-digits problem.

Targeted Maximum Likelihood Learning (laan-2006-target-maxim)

See koenker-2016-tmle for a good tutorial, as well as this slide deck for Stata: Ensemble Learning Targeted Maximum Likelihood Estimation for Stata Users.

Ten quick tips for machine learning in computational biology (chicco-2017-ten-quick)

1. Check and arrange your input dataset properly
2. Split your input dataset into three independent subsets (training set, validation set, test set), and use the test set only once you complete training and optimization phases
3. Frame your biological problem into the right algorithm category
4. Which algorithm should you choose to start? The simplest one!
5. Take care of the imbalanced data problem
6. Optimize each hyper-parameter
7. Minimize overfitting
8. Evaluate your algorithm performance with the Matthews correlation coefficient (MCC) or the Precision-Recall curve
9. Program your software with open source code and platforms
10. Ask for feedback and help to computer science experts, or to collaborative Q&A online communities

The elements of programming style (kernighan-1978-elemen-progr-style)

Nice introductory example to build an identity matrix in Fortran, which however would read much better using simple imperative code.

> Write clearly – don't be too clever.

A related advice is "write clearly – don't sacrifice clarity for 'efficiency'."

An n-by-n matrix has n^2 elements, which means n^2 assignments for its initialization. Multiplying two such matrices or solving n linear equations of n unknowns require on the order of n^3 operations. If n >= 10, the time required to initialize a matrix is not important. What if n < 10? (Hint: I/O operations are more time consuming than arithmetic.)

All rules are listed at the end (pp. 159-161).

The Imposter's Handbook (conery-2016-impos-handb)

- review published on aliquote.org
- Source code on Github (JS, C#, Bash, SQL)

The Little Schemer (friedman-1995-littl-schem)

Beautiful book, very different from SICP in that it focus on basic building blocks (`car`, `cdr`, `cons`, `eq?`, etc.) and use a very pragmatic approach to understanding the structuration and interpretation of forms and s-expr. The penultimate goal of this book (4th ed., after the original *Little Lisper*) is to learn to think in a functional way. The ten commandments are worth keeping in mind for that very specific purpose:

1. When recurring on a list of atoms, `lat`, ask two questions about it: `(null? lat)` and `else`. When recurring on a number, `n`, ask two questions about it: `(zero? n)` and `else`. When recurring on a list of s-expr, `l`, ask three questions about it: `(null? l)`, `(atom? (car l))`, and `else`.
2. Use `cons` to build lists.
3. When building a list, describe the first typical element, and then `cons` it into the natural recursion.
4. Always change at least one argument while recurring. When recurring on a list of atoms, `lat`, use `(cdr lat)`. When recurring on a number, `n`, use `(sub1 n)`. And when recurring on a list of s-expr, `l`, use `(car l)` and `(cdr l)` if neither `(null? l)` nor `(atom? (car l))` are true. It must be changed to be closer to termination. The changing argument must be tested in the termination condition: when using `cdr`, test termination with `null?`, and when using `sub1`, test termination with `zero?`.
5. When building a value with ÷, always use 0 for the value of the terminating line, for adding 0 does not change the value of an addition. When building a value with x, always use 1 for the value of the terminating line, for multiplying by 1 does not change the value of a multiplication. When building a value with `cons`, always consider `()` for the value of the terminating line.
6. Simplify only after the function is correct.
7. Recur on the subparts that are of the same nature:
   - on the sublists of a list;
   - on the subexpressions of an arithmetic expression.

8. Use help functions to abstract from representations.
9. Abstract common patterns with a new function.
10. Build functions to collect more than one value at a time.

The Probable Error of a Mean (gosset-1908-probab-error-mean)

Extra R code (Frank Harrell, harrell-2017-biost-biomed-resear)

```
drug1 = c(0.7, -1.6, -0.2, -1.2, -0.1, 3.4, 3.7, 0.8, 0, 2)
drug2 = c(1.9, 0.8, 1.1, 0.1, -0.1, 4.4, 5.5, 1.6, 4.6, 3.4)
d = data.frame(Drug=c(rep('Drug 1', 10), rep('Drug 2', 10), rep('Difference', 10)),
               extra=c(drug1 , drug2 , drug2 - drug1))
w = data.frame(drug1, drug2, diff=drug2 - drug1)
ggplot(d, aes(x=Drug, y=extra)) +
geom_boxplot(col='lightyellow1', alpha=.3, width=.5) +
geom_dotplot(binaxis='y', stackdir='center', position='dodge') +
stat_summary(fun.y=mean, geom="point", col='red', shape=18, size=5) +
geom_segment(data=w, aes(x='Drug 1', xend='Drug 2', y=drug1, yend=drug2), col=gray(.8)) +
geom_segment(data=w, aes(x='Drug 1', xend='Difference', y=drug1, yend=drug2 - drug1), col=gray(.8)) +
xlab('') + ylab('Extra Hours of Sleep') + coord_flip()
```

The weak spots in contemporary science (and how to fix them) (wicherts-2017-weak-spots)

Objectives: demonstrate that the pluridisciplinar crisis in science can mainly be accounted for by observer bias, publication bias, misuse of degrees of freedom in statistical analysis of data combined to low statistical power, and errors in the reporting of results.

Up to 90% of positive results reported in psychology or psychiatry.

HARKing: *Hypothesizing after Results are Known*—much like "data fishing", or to a lesser extent "data dredging".

Ioannidis's work on reproductibility and misuse of statistical hypothesis testing framework: ioannidis-2005-why-most, munafo-2017-manif-reprod-scien.

Theoretical impediments to machine learning with seven sparks from the causal revolution (pearl-2018-theor-imped)

Seven tasks which are beyond reach of current machine learning systems (vs. structural causal models) and examples of tasks ML would fail to solve: (1) How effective is a given treatment in preventing a disease?, (2) Was it the new tax break that caused our sales to go up?, (3) What is the annual health-care costs attributed to obesity?, (4) Can hiring records prove an employer guilty of sex discrimination?, (5) I am about to quit my gob, but should I?

Topological Data Analysis (wasserman-2016-topol-data-analy)

Topological data analysis = finding structure in data, i.e., clustering, manifold estimation, nonlinear dimension reduction, mode estimation, ridge estimation and persistent homology. The latter is often what people understand when we talk about topological data analysis. The author extends the notion a bit, but does not discuss shape manifolds. There is another field that deals with the topological and geometric structure of data: computational geometry. The main difference is that in TDA we treat the data as random points whereas in computational geometry the data are usually seen as fixed.

See also the R package TDA.

Twenty years of attacks on the rsa cryptosystem (boneh-2002-twent-years)

There are many Coppersmith-based attacks, but this mostly resolves around the case where public exponent e is small or when partial knowledge of the secret key is available:

- **Small decryption exponent e:** so far the best known attack recovers e if it is less than $N^{.292}$. This uses a bivariate version of Coppersmith that lacks a rigorous proof of correctness, but seems to work well in practice. Important open questions are whether $e < N^1/2$-e is attackable (the conjecture is that it should be), and whether there are rigorously provable variants of Coppersmith for bivariate or multivariate polynomials.
- **Partial secret key exposure:** when certain bits of e or the factors p, q of N are exposed, it is often possible to recover them completely.

Using reference models in variable selection (pavone-2020-using)

A reference model is used to mimic the data-generating process, under the assumption of an M-complete framework whereby such a model reflects our beliefs about the future data in the best possible way and has passed model checking and criticism. When using a bayesian reference model, the idea is to project its predictive distribution to a reduced model leading to projection predictive variable selection approach. See also `http://mc-stan.org/projpred`. Side note: Now I know that what I've been using 8 years ago for variable selection using rerandomization or bootstrap may be called something like "Bootstrap inclusion frequencies" (see Fig. 1).

What is a statistical model (mccullagh-2002-what-statis-model)

From John D Cook's blog.

The author suggests that "most authors do not offer a precise mathematical definition of a statistical model", and gives 12 examples of ill-posed statitsical models from an inferential perspective.

Starting page 1232 ff., it is all about category theory!

> The thesis of this paper is that the logic of every statistical model is founded, implicitly or explicitly, on categories of morphisms of the relevant spaces. The purpose of a category is to ensure that the families of distributions on different sample spaces are logically related to one another and to ensure that the meaning of a parameter is retained from one family to another.

What is category theory (bradley-2018-what-categ-theor)

- Main blog
- Level: graduate student

Category Theory used to reshape and reformulate problems within pure mathematics, including topology, homotopy theory and algebraic geometry, and it has various applications in *chemistry*, neuroscience, systems biology, *natural language processing*, causality, network theory, dynamical systems, and database theory.

Two central themes:

- functorial semantics: C $\to$ D ~ interpretation of C within D; syntax (grammar in NLP) refers to rules for putting things together and semantics (meaning) refers to the meaning of those things.
- compositionality

Why Rust? (blandy-2015-why-rust)

Rust, like Python, JS or Ruby, is a type safe language with immutable variables by default, but it also allows the use of `unsafe` code and `mut` able variables. Moreover, "Rust's particular form of type safety guarantees that concurrent code is free of data races, catching any misuse of mutexes or other synchronization primitives at compile time, and permitting a much less adversarial stance towards exploiting parallelism." In addition, Rust guarantees memory safety through three key promises: no null pointer dereferences, no dangling pointers and no buffer overruns.

Rust offers a flexible macro system (not covered in this short review); see the official documentation or the Rust by Example. There are also *generic* types and functions, like C++ templates, except that in Rust we must specifiy the type of the argument `T` (`Ord` in the example below):

```
fn min<T: Ord>(a: T, b: T) -> T {
  if a <= b { a } else { b }
}
```

Note that "Rust compiles generic functions by producing a copy of their code specialized for the exact types they're applied to."

Rust enumerated types can be viewed as kind of *algebric datatypes* (equivalent to "tagged union" in C):

```
enum Option<T> {
  None,
  Some(T)
}

fn safe_div(n: i32, d: i32) -> Option<i32> {
  if d == 0 {
    return None;
  }
  return Some(n / d);
}

// We need to check either variant of the enumerated type
match safe_div(num, denom) {
        None => println!("No quotient."),
        Some(v) => println!("quotient is {}", v)
}
```

See other examples of use regarding memory safety.

Iterators and traits, the later being a "collection of functionality that a type can implement"), pp. 11-17.

```
// https://stackoverflow.com/a/45283083
// Iterators are lazy and process each element only once.
fn main() {
  let v1 = (0u32..9).filter(|x| x % 2 == 0).map(|x| x.pow(2)).collect::<Vec<_>>();
  let v2 = (1..10).filter(|x| x % 2 == 0).collect::<Vec<u32>>();

  println!("{:?}", v1);
  println!("{:?}", v2);
}
```

Some additional pointers:

- Rust book: The Rust Programming Language
- Evan Miller's review: A Taste of Rust
- Jeroen Ooms (@opencpu): Hello Rust (Minimal Example of Calling Rust from R using Cargo)

Why you cannot (yet) write an "interval arithmetic" library in common lisp (antoniotti-2020-why-you)

See also this blog post.

Calculating the sample size required for developing a clinical prediction model (riley-2020-calcul)

Fine distinction between the 10 EPV and 10 EPP rules of thumb: it's important to consider the total number of predictor parameters considered and not variables (which may have several associated β parameters) or parameters included in the final model. The very inconsistent recommendations about EPP suggest that it is actually context specific and depends not only on the number of events relative to the number of candidate predictor parameters but also on the total number of participants, the outcome proportion (incidence) in the study population, and the expected predictive performance of the model.

Recall what Frank Harrell advocates for logistic regression (n=96 to assess the intercept only):

> A simple way to do this is to calculate the sample size needed to precisely estimate (within a small margin of error) the intercept in a model when no predictors are included (the null model).

This yields the following estimate (when aiming at computing a precise estimate of the overall outcome risk): With a binary outcome that occurs in half of individuals, a sample size of at least 385 people is needed to target a confidence interval of 0.45 to 0.55 for the overall outcome proportion, and thus an error of at most 0.05 around the true value of 0.5. To achieve the same margin of error with outcome proportions of 0.1 and 0.2, at least 139 and 246 participants, respectively, are required. For time-to-event outcomes, a key time point needs to be identified, along with the anticipated outcome event rate. For example, with an anticipated event rate of 10 per 100 person years of the entire follow-up, the sample size must include a total of 2366 person years of follow-up to ensure an expected margin of error less than 0.05 in the estimate of a 10 year outcome probability of 0.63, such that the expected confidence interval is 0.58 to 0.68. See Fig. 1 and Box 1.

On the design of text editors (rougier-2020-desig-text-editor)

Elegant Emacs available on Github. Nice discussion about the use and abuse of syntax highlighting ("Christmas tree effect"). Figure 4 reminds me of code snippets I found on Vincent Zoonekynd's website but which are no longer available apparently.

Why Functional Programming Matters (hughes-1990-why-funct)

Key idea : Two features of functional languages in particular, higher-order functions and lazy evaluation, can contribute significantly to modularity. As an example of higher-order function, the author provides extensive illustration of the use of `foldr` to compute the sum, or any other mathematical constructs, of the elements of an arbitrary list. Trees and the Newton-Raphson algorithm are also discussed in the context of gluing functions together, i.e. `(g . f)`, as well as other numerical algorithms (differentiation and integration). Finally, the tic-tac-toe game is used to demonstrate how dynamic programs can benefit from lazy evaluation.

> Functional programmers argue that there are great material benefits — that a functional programmer is an order of magnitude more productive than his or her conventional counterpart, because functional programs are an order of magnitude shorter. Yet why should this be? The only faintly plausible reason one can suggest on the basis of these "advantages" is that conventional programs consist of 90% assignment statements, and in functional programs these can be omitted! This is plainly ridiculous. If omitting assignment statements brought such enormous benefits then Fortran programmers would have been doing it for twenty years. It is a logical impossibility to make a language more powerful by omitting features, no matter how bad they may be.

Une introduction agreable au langage haskell 98 (hudak-2007-une-haskel)

Very interesting discussion around Haskell typing system (section II). Right (e.g., list append) and left (e.g., function application) associativity are tricky.

```
inc :: Integer -> Integer
inc n = n+1

add :: Integer -> Integer -> Integer
add x y = x + y
```

In the above example, the redefinition of `inc` as `inc = add 1` is an example of partial application of a curried function. Here is a more complex example:

```
map :: (a->b) -> [a] -> [b]
map f [] = []
map f (x :xs) = f x : map f xs
```

Which monads haskell developers use: an exploratory study (figueroa-2021-which-haskel)

The authors use the Hasckage central package and `.cabal` files to study how many packages rely on the Monad Transformers Library (as imported or provided modules, or as dependencies), which monads are currently in use, and whether alternative implementation of monads are used. Regarding the use of MTL, results show that there are close to 10,000 packages involving at least 1 monad, and the most frequence monad are `State`, `Reader` and `Trans`.

Functional data structures for typed racket (prashanth-2010-funct-data)

Example of builtin functional data structure in Scheme: the linked list. Remember that lists are usually mutable in Scheme. Other efficient data structures proposed by @okasaki-1996-purel-funct or @bagwell-2001-ideal-hash-trees are available in Haskell or Clojure. The authors describe several higher-order data structure implemented in Typed Racket: queue, deque, heaps, extensions on lists (random access, streams, hash, etc.).

See also:

Phil Bagwell. Fast Functional Lists, Hash-Lists, Deques and Variable Length Arrays. In Implementation of Functional Languages, 14th In- ternational Workshop, 2002.

Evolution of emacs lisp (monnier-2020-evolut-emacs-lisp)

History: TECO < Emacs (EINE, then ZWEI) < Multics Emacs < Unix (Gosling, then Unipress) Emacs < GNU Emacs, then its fork, Lucid Emacs (later XEmacs). Emacs 13 (1985, Richard Stallman) -> Emacs 26.1 (2018, John Wiegley and Eli Zaretskii).

The Zipper (huet-1997-zipper)

On the use of destructive operations in data structures. Basic idea: the tree is turned inside-out like a returned glove, pointers from the root to the current position being reversed in a path structure. The current location holds both the downward current subtree and the upward path. All navigation and modification primitives operate on the location structure. Going up and down in the structure is analogous to closing and opening a zipper in a piece of clothing, whence the name.

Complete Ocaml implementation is provided.

> Efficient destructive algorithms on binary trees may be programmed with these completely applicative primitives, which all use constant

time, since they all reduce to local pointer manipulation.

Tidy Data (wickham-2014-tidy-data)

The core idea of a tidy dataset is that "each variable is a column, each observation is a row, and each type of observational unit is a table". This should sound familiar to people used to SQL, database normalization, and OLAP principles.

This idea became the baiss of the tidyr package, superseding the `reshape2` package (used to switch from wide to long format, and the converse), and closely related to the tibble or tribble concept of a data table (which is different from what offer standard data frames or the data.table package).

The split-apply-combine strategy for data analysis (wickham-2011-split-apply)

It was a nice paper at the time, and I used it a lot to illustrate ideas related to data management and data cleansing.

The plyr[3] package was great, with some idiosyncratic notation for arguments (e.g., `.data`). It allowed to return multple values which was possible but trickier with `Hmisc` and unlikely to be useful with `tapply` or `aggregate`.

[3] Now plyr refers to simple, accessible and customisable media player

> (...) break up a big problem into manageable pieces, operate on each piece independently and then put all the pieces back together.

Base R versus `plyr`:

#+BEGIN$_{SRC}$ R ## base spl = with(crabs, split(BD, sp)) apl = lapply(spl, mean) cbn = rbind(x = apl) cbn ## plyr plyr::ddply(data = crabs, .(sp), summarize, x = mean(BD)) ## plyr::ddply(crabs, "sp", summarize, x = mean(BD)) #+END$_{SRC}$=

Later it has been supplnated by `dplyr`, and later on by a combination of `dplyr` + `tidyr`, and what has now become the "tidyverse" (`dplyr` + `tidyr` + `forcats` + whatever).

Topology and data (carlsson-2009-topol-data)

point clouds = finite sets of points (representing finite and possibly noisy samples taken from a geometric object) with an associated distance function. The concept of metrics and coordinates are often ill-defined in a biological context (e.g. BLAST scores).

> One method of clustering a point cloud is the so-called single linkage clustering, in which a graph is constructed whose vertex set is the set of points in the cloud, and where two such points are connected by an edge if their distance is <= e, where e is a parameter. Some

> work in clustering theory has been done in trying to determine the optimal choice of e, but it is now well understood that it is much more informative to maintain the entire dendrogram of the set, which provides a summary of the behavior of clustering under all possible values of the parameter e at once. It is therefore productive to develop other mechanisms in which the behavior of invariants or construction under a change of parameters can be effectively summarized.

Topology "includes the study of what the connected components of a space are, but more generally it is the study of connectivity information, which includes the classification of loops and higher dimensional surfaces within the space."

TODO: Draw some connections with Category Theory.

A Scheme concurrency library (kato-2017-schem)

Description of a concurrent library built on top of SRFI-18. Implementation of (1) shared queue as a record cith usual queue structure fields (mutex and condition variable), and (2) thread pool as a container which holds specified number of threads which never stop until stop operation is explicitly invoked.

A Literate Program (bentley-1986-liter-progr)

The famous paper where Jon Bentley devised a short Unix pipeline while Don Knuth wrote a full-length web program to solve the following task: Given a text file and an integer k, print the k most common words in the file (and the number of their occurrences) in decreasing frequency.

```
tr -cs A-Za-z' ' | tr A-Z a-z | sort | uniq -c | sort -rn | sed ${1}q
```

Strong Inference (platt-1964-stron-infer)

Canonical paper where Platt argues for the need of devising alternative hypothesis and reliable experiment (with alternative possible outcomes) instead of focusing on a single hypothesis to avoid confirmation biais. This is mostly a scientific method of thinking, not a statistical framework, but in a certain sense the likelihood paradigm is probably the best framework for that purpose as it allows to compare alternative models.

On Abandoning Xlisp-Stat (de-2005-aband-xlisp-stat)

This article led me to revisit Lispstat, the Lisp-related code found on UCLA servers (especially the ones written by Jan de Leeuw in the 80s), and my interest in Lisp-based statistical packages grew up with Ross Ihaka's paper (@ihaka-2008-back-futur).

Back to the future: lisp as a base for a statistical computing system (ihaka-2008-back-futur)

There was no real follow-up in terms of a new language design, JIT compiler or alternative to R. On the one hand, Tony Rossini worked on a reimplementation of LispStat in Common Lisp but the development has staled quickly. On the other hand, there was already Clojure (first with Incanter, then with third-party libs) and later Julia came out, and they took the lead on modern software for numerical and statistical computing.

See also: reddit thread.

False discovery rates: a new deal (stephens-2017-false)

Unlike traditional FDR computation, the author suggest to use effect size and its standard error as input to the local FDR[4] computation, which yields interval estimates (credible regions) for each effect in addition to measures of significance.

Gelman's type S errors are handled using the local false sign rate, following Tukey recommendation to answer the question "is the evidence strong enough to support a belief that the observed difference has the correct sign?" rather than "are the effects different?".

See also: R package ashr.

[4] The local FDR is the probability, given the observed data, that effect j would be a false discovery, if we were to declare it a discovery.

Teaching stats for data science (kaplan-2017-teach)

This article provides a nice overview of key concepts to teach to people interested in Data Science. Not sure the term coined so long ago (Cleveland, 2001) but over-emphasized in recent years will last, but that's another story.

> The operations to be applied each reflect a different part of the process of statistical thinking: modeling is about constructing a description from data, effect size is about summarizing that description, bootstrapping addresses the issue of precision, and cross-validation informs choices about what explanatory variables or model architectures should be used.

See also donoho-2017-years-data-scien and `https://aliquote.org/post/50-years-data-science/`.

Using functions for easier programming (savage-2018-using-funct)

Short review (mostly citations by known programmers or language developers, e.g. Simon Peyton Jones) of the advantage of functional programing over imperative or OO approaches (which mostly resolves around object state and mutability).

Best practices for scientific computing (wilson-2014-best-pract)

Summary of best practices, from the Software Carpentry team:

1. Write programs for people, not computers.
2. Let the computer do the work.
3. Make incremental changes.
4. Don't repeat yourself (or others).
5. Plan for mistakes.
6. Optimize software only after it works correctly.
7. Document design and purpose, not mechanics.
8. Collaborate.